CEUR-WS.org/Vol-1176
urn:nbn:de:0074-1176-8

CLEF 2010
CLEF2010 Working Notes

Working Notes for CLEF 2010 Conference

Padua, Italy, September 22-23, 2010.

Edited by

Martin Braschler *
Donna Harman **
Emanuele Pianta
Nicola Ferro (CEUR-WS Republishing Managing Editor) ***

* Zurich University of Applied Sciences, Zurich, Switzerland
** NIST, National Institute of Standards and Technology, U.S.A.
*** Department of Information Engineering (DEI), University of Padua, Via Gradenigo 6/B, 35131, Padova, Italy

Table of Contents

Preface

Cross-Language Retrieval in Image Collections

- Visual Localization Using Global Visual Features and Vanishing Points
  Olivier Saurer, Friedrich Fraundorfer, Marc Pollefeys
- Meiji University at ImageCLEF2010 Visual Concept Detection and Annotation Task
  Naoki Motohashi, Ryo Izawa, Tomohiro Takagi
- A Text-Based Approach to the ImageCLEF 2010 Photo Annotation Task
  Wei Li, Jinming Min, Gareth J. F. Jones
- A Novel Structural-Description Approach for Image Retrieval
  Christoph Rasche, Constantin Vertan
- LEAR and XRCE's Participation to Visual Concept Detection Task - ImageCLEF 2010
  Thomas Mensink, Gabriela Csurka, Florent Perronnin, Jorge Sánchez, Jakob Verbeek
- MRIM-LIG at ImageCLEF 2010 Visual Concept Detection and Annotation task
  Rami Al Batal, Philippe Mulhem
- TELECOM ParisTech at ImageCLEF 2010 Photo Annotation Task: Combining Tags and Visual Features for Learning-Based Image Annotation
  Hichem Sahbi, Xi Li
- Detection of Visual Concepts and Annotation of Images Using Predictive Clustering Trees
  Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, Sašo Džeroski
- UPMC/LIP6 at ImageCLEFannotation 2010
  Ali Fakeri-Tabrizi, Sabrina Tollari, Nicolas Usunier, Massih-Reza Amini, Patrick Gallinari
- Random Sampling Image to Class Distance for Photo Annotation
  Deyuan Zhang, Bingquan Liu, Chengjie Sun, Xiaolong Wang
- Augmenting Bag-of-Words - Category Specific Features and Concept Reasoning
  Eugene Mbanya, Christian Hentschel, Sebastian Gerke, Mohan Liu, Andreas Nürnberger, Patrick Ndjiki-Nya
- The Wroclaw University of Technology Participation at ImageCLEF 2010 Photo Annotation Track
  Michal Stanek, Oskar Maier, Halina Kwasnicka
- Linear SVM for new Pyramidal Multi-Level Visual only Concept Detection in CLEF 2010 Challenge
  Sébastien Paris, Hervé Glotin
- The University of Amsterdam's Concept Detection System at ImageCLEF 2010
  Koen E. A. van de Sande, Theo Gevers

Cross-lingual Expert Search Workshop

- Overview of the Cross-lingual Expert Search (CriES) Pilot Challenge
  Philipp Sorg, Philipp Cimiano, Antje Schultz, Sergej Sizov
- HITS and Misses: Combining BM25 with HITS for Expert Search
  Johannes Leveling, Gareth J. F. Jones

- Multilingual Expert Search using Linked Open Data as Interlingual Representation
  Daniel M. Herzig, Hristina Taneva
- Identify Experts from a Domain of Interest
  Adrian Iftene, Bogdan Luca, Georgiana Cărăuşu, Madălina Merchez

Log File Analysis

- LogCLEF 2010: the CLEF 2010 Multilingual Logfile Analysis Track Overview
  Thomas Mandl, Giorgio Maria Di Nunzio, Julia Maria Schulz
- Ambiguity of Queries and the Challenges for Query Language Detection
  Juliane Stiller, Maria Gäde, Vivien Petras
- How does the Library Searcher Behave? A Contrastive Study of Library Search against Ad-hoc Search
  Suzan Verberne, Max Hinne, Maarten van der Heijden, Eduard Hoenkamp, Wessel Kraaij, Theo van der Weide
- SINAI at LogCLEF 2010
  José Manuel Peréa-Ortega, Arturo Montejo-Ráez, Miguel Angel García-Cumbreras , L. Alfonso Ureña-López
- Language Identification Strategies for Cross Language Information Retrieval
  Alessio Bosca, Luca Dini
- DCU-TCD@LogCLEF 2010: Re-ranking Document Collections and Query Performance Estimation
  Johannes Leveling, M. Rami Ghorab, Walid Magdy, Gareth J. F. Jones, Vincent Wade
- DAEDALUS at LogCLEF 2010: Analyzing the Success of Search Queries
  Sara Lana-Serrano, Julio Villena-Román, José Carlos González-Cristóbal
- CRES at LogCLEF 2010: Towards Understanding the User Behaviors through an Analysis of Search Sessions, Search Units and Click Ranks
  Masao Takaku, Yuka Egusa, Hitomi Saito, Noriko Kando, Hitoshi Terai, Makiko Miwa

Multiple Language Question Answering 2010

- Overview of ResPubliQA 2010: Question Answering Evaluation over European Legislation
  Anselmo Peñas, Pamela Forner, Álvaro Rodrigo, Richard Sutcliffe, Corina Forăscu, Cristina Mota
- Question Answering System: Retrieving Relevant Passages
  Hitesh Sabnani, Prasenjit Majumder
- JU_CSE_TE: System Description QA@CLEF 2010 - ResPubliQA
  Partha Pakray, Pinaki Bhaskar, Santanu Pal, Dipankar Das, Sivaji Bandyopadhyay, Alexander Gelbukh
- FIDJI @ ResPubliQA 2010
  Xavier Tannier, Véronique Moriceau

Retrieval in the Intellectual Property Domain

Uncovering plagiarism, authorship and social software misuse

2010
Rafael Corezola Pereira, Viviane P. Moreira, Renata Galante

- Exploring Fingerprinting as External Plagiarism Detection Method - Lab Report for PAN at CLEF 2010
  Yurii Palkovskii, Alexei Belov, Irina Muzika
- External Plagiarism Detection - Lab Report for PAN at CLEF 2010
  Sobha Lalitha Devi, Pattabhi R. K. Rao, Vijay Sundar Ram, A. Akilandeswari
- External Plagiarism Detection Based on Standard IR Technology and Fast Recognition of Common Subsequences - Lab Report for PAN at CLEF 2010
  Thomas Gottron
- A Textual-Based Similarity Approach for Efficient and Scalable External Plagiarism Analysis - Lab Report for PAN at CLEF 2010
  Daniel Micol, Óscar Ferrández, Fernando Llopis, Rafael Muñoz
- Plagiarism Detection Using Information Retrieval and Similarity Measures Based on Image Processing Techniques - Lab Report for PAN at CLEF 2010
  Marta R. Costa-jussà, Rafael E. Banchs, Jens Grivolla, Joan Codina
- University of Sheffield - Lab Report for PAN at CLEF 2010
  Rao Muhammad Adeel Nawab, Mark Stevenson, Paul Clough
- External Plagiarism Detection: N-Gram Approach Using Named Entity Recognizer - Lab Report for PAN at CLEF 2010
  Parth Gupta, Sameer Rao, Prasenjit Majumder
- Automatic External Plagiarism Detection Using Passage Similarities - Lab Report for PAN at CLEF 2010
  Clara Vania, Mirna Adriani
- A Plagiarism Detector for Intrinsic Plagiarism - Lab Report for PAN at CLEF 2010
  Pablo Suárez, José Carlos González-Cristóbal, Julio Villena-Román
- Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection - Lab Report for PAN at CLEF 2010
  Salha Alzahrani, Naomie Salim
- Wikipedia Vandalism Detection Through Machine Learning: Feature Review and New Proposals - Lab Report for PAN at CLEF 2010
  Santiago Moisés Mola Velasco
- Detecting Wikipedia Vandalism using WikiTrust - Lab Report for PAN at CLEF 2010
  B. Thomas Adler, Luca de Alfaro, Ian Pye
- Novel Balanced Feature Representation for Wikipedia Vandalism Detection Task - Lab Report for PAN at CLEF 2010
  István Hegedũs, Róbert Ormándi, Richárd Farkas, Márk Jelasity
- Wiki Vandalysis - Wikipedia Vandalism Analysis - Lab Report for PAN at CLEF 2010
  Manoj Harpalani, Thanadit Phumprao, Megha Bassi, Michael Hart, Rob Johnson
- ZOT! to Wikipedia Vandalism - Lab Report for PAN at CLEF 2010
  James White, Rebecca Maessen

Web People Search

2014-08-11: submitted by Nicola Ferro
2014-08-24: published on CEUR-WS.org |valid HTML5|

Contextual Approach for Paragraph Selection in Question Answering Task

Hapnes Toba, Syandra Sari, Mirna Adriani, and Ruli Manurung

Information Retrieval Laboratory, Faculty of Computer Science, University of Indonesia
Depok Campus, Indonesia
hapnes.toba@ui.ac.id, syandra.sari@ui.ac.id, mirna@cs.ui.ac.id, maruli@cs.ui.ac.id

Abstract. This year we participated in the English monolingual paragraph selection task at ResPubliQA 2010. Our general strategy is to find the supporting word context from query and candidate passages during the paragraph selection. We use the techniques of state-of-the art publicly available question answering systems, i.e. Open Ephyra and JIRS, and the random projection implementation in the Semantic Vectors package to evaluate the word context. To strengthen the paragraph selection, besides the context evaluation, we also use *n*-gram overlapping and textual containment. Our approach has a c@1 measure of 0.73 for our pattern-based context configuration and 0.64 for our *n*-gram-based context configuration.

Keywords: context supporting, passage retrieval, random projection, *n*-gram overlapping, textual containment.

1 Introduction

Question Answering (QA) is a specific form of information retrieval (IR) that seeks to produce an exact answer given a natural language question. An automated QA system tries to retrieve explicit answers in the form of a single answer or snippets of text rather than a whole document or set of documents. The main techniques that mostly have been used in current QA research are semantic analysis using semantic role labeling [20], named entity recognition [5], path dependency [5], semantic markup [12], *n*-gram passages [2, 7, 11], statistical methods [13, 14], combinations of semantic structures and probabilistic approaches [15], and combinations of semantic structures and automated reasoning [16, 17, 18]. There is no ultimate technique, each approach has its own role, application domain, and tasks [19].

This year's ResPubliQA[1] evaluation campaign is a continuation from last year which tries to evaluate QA performance in a specific context, i.e. the legal domain [21]. Since this year is our first participation, we decided to compete in the English monolingual paragraph selection task. We experiment with various QA strategies that are supported by the notion of word context, which return passages that contain candidate answers. For this purpose, we develop a context supporting paragraph

[1] http://celct.isti.cnr.it/ResPubliQA/

selection strategy which validates the passage retrieval mechanisms from two state-of-the-art publicly available QA systems, Open Ephyra [1] and JIRS [2, 7].

Our general hypothesis in this experiment is that, in a specific domain, the context of a question is near to the context of candidate paragraphs that are suggested during the passage retrieval. This hypothesis is a generalization of the distributional hypothesis from the word space methodology, which says that words with similar meaning tend to occur in similar contexts [3]. We took the publicly available random projection implementation in Semantic Vectors [4] as the context supporting system of our approach. Our observation during this experiment will be: how to employ word context to support passage retrieval performed by language model and *n*-gram approaches. To answer this research question, our approach tries to boost the influence of word context during paragraph selection.

In this paper we describe the approach of our context supporting strategy and the results obtained in the monolingual English paragraph selection task. The rest of the paper is structured as follows: in section 2 we describe in general the related systems whose techniques were used in our approach, namely Open Ephyra, JIRS and Semantic Vectors. The details of our approach will be covered in section 3. Section 4 will show some results and analysis in the paragraph selection task of this year's ResPubliQA. Finally, some conclusions and suggestions for future work will be given in section 5.

2 QA Pipeline and Word Space Methodology

Open Ephyra[2] and JIRS[3] are two QA systems which offer comprehensive pipelines. They base their passage retrieval strategies on, respectively, language models and *n*-gram structures of the passages. Both strategies rely on the probability and sequences of adjacent words from query-passage pairs, but do not really observe the word in context. Alternatively, Semantic Vectors[4] offers the possibility to observe the word context in a domain by implementing the word space methodology.

2.1 Pattern-based Question Answering

Open Ephyra (OE) uses a pattern learning approach to categorize questions [1, 5]. OE can learn question-answer pairs and uses standard IR systems, such as Indri, to fetch text snippets that are suitable for pattern extraction. It consists of four main modules: a question analyzer, query generator, search engine, and answer extractor. Each module can be used independently and is thus suitable for experimenting with multiple approaches to question-answering as one pipeline system.

There are two main steps for the pattern-learning approach in OE:

1. The first is to learn the question patterns from question templates according to each question type. The aim of this step is to interpret the questions and

[2] http://www.ephyra.info
[3] http://sourceforge.net/projects/jirs/
[4] http://semanticvectors.googlecode.com/

transform them into queries. The question templates need to be manually developed according to various interrogative sentences that are independent for each natural language.

2. The second step is to learn the answer patterns from question-answer pairs. The aim of this second step is to extract answer candidates from relevant document snippets and to rank them.

In our experiment, we only used the first step to develop an appropriate set of question patterns for each question type according to the training data from last year's ResPubliQA. A recognized question pattern for a given question is essential in order to extract important keywords (i.e. target, context and property) for querying purposes. The retrieval phase in OE is done within the Indri search engine that searches for passages in documents - based on the recognized query keywords - which could contain the answer candidates.

2.2 *n*-Gram-based Question Answering

The question answering module in JIRS is an extension of the JIRS passage retrieval system as described in [7]. The general architecture of JIRS is comparable to OE as described in the previous sub-section. A given question will be classified to an appropriate class that will be further used by the passage retrieval algorithm. JIRS's retrieval algorithm is based on the ordering of *n*- neighboring words extracted from a passage, which is called as Clustered Keyword Positional Distance (CKPD).

JIRS is based on the idea that in a large document collection, an *n*-gram related with a question will be found in the collection at least once [11]. Only passages with *n*-grams that contain question terms are returned. The weight of each passage is calculated according to the similarity between the question and the passage n-grams. The similarity of a passage with the question is greater if the passage shares longer structures with the question [2]. A brief introduction of JIRS can found in [11], while the complete CKPD algorithm can be found in [2].

2.3 Semantic Vectors

Semantic Vectors is an open source package that can be used to build context vectors of word concepts in a specific domain. It implements word space methodology [3] by applying random projections of words in the document collection. Words in Semantic Vectors are represented as vectors in a high dimensional space, where words or documents that have related meanings are in close proximity. By applying random projection, the computational resources that are usually required for computing semantic similarity, as for instance in Latent Semantic Analysis (LSA), can be minimized.

The random projection strategy can be summarized in two main steps [3, 4, 8]:

1. Step 1: Build the term vectors.
   For each document, make an *N*-dimensional ternary index vector by placing a small number, *k*, of -1's and +1's (e.g., ten of each; *k* = 10) at random among the *N*-dimensional index vector, the rest will be 0's.

2. Step 2: Build the document vectors
   Scan through the documents. Every time the word appears, add the index vector to row *w* of matrix *G*, where *G* is a (*M* x *N*) matrix, *M* is the number of words in the collection and *N* is the number of reduced columns.

The Semantic Vectors package uses the Apache Lucene API[5] to create a word space model from a term document matrix, using random projection to perform dimensionality reduction.

3 Experiments

3.1 Data Preparation

For the data preparation, from the JRC-ACQUIS[6] and EUROPARL[7] corpus we have created a passage index based on the paragraph segmentations. In total we have around 1.5 million passages. We created separate indexes for each information retrieval system that we used in our approach, namely Indri, JIRS and Lucene.

Indri is a search engine that is specially designed for passage retrieval [2] such as JIRS. The difference between them lies in the retrieval model. Indri's retrieval model is based on a combination of language modeling and inference network retrieval frameworks [6], while JIRS based its retrieval model on the CKPD algorithm [2].

The Lucene index will be used by the Semantic Vectors package to form the terms and document vectors [4]. In our approach we choose the value of 2000 for the number of dimensions. This value is the reduced columns for documents projection that is considered as the word space.

3.2 System Architecture

The high level architecture of our approach can be seen in Fig. 1. This architecture is based on the general framework of question answering systems. The first component is the question analysis, which normalizes the question (i.e. removes the question word, stop-words, and performs word stemming), and determines the question type. We used manually developed question patterns to determine the question types.

After the question analysis step, the system delivers the normalized question to a query generator based on the techniques that were used in OE [1, 5], JIRS [2] and Semantic Vectors [4]. The next step is the information retrieval phase. In our approach, it is done by the Indri, JIRS or Lucene search engine, which corresponds to the generated queries. The retrieval results from each search engine are collected in separated files that will be evaluated by the paragraph selection component.

The main idea during paragraph selection is to evaluate whether supporting word context can be found among the candidate passages. It checks first the word context

[5] http://lucene.apache.org/

[6] http://wt.jrc.it/lt/Acquis/

[7] http://www.europarl.europa.eu/

that is present at the top-1 retrieval of the candidate passages. This is done by comparing candidate passages from Indri or JIRS with the one retrieved by Semantic Vectors. If the two systems that were compared returned the same document at the top-1 retrieval, then we believe that supporting word(s) context was found in the passages that were compared, or in other words the documents “shared” the same context. Otherwise, the paragraph selection will be determined by applying *n*-gram overlapping, textual containment [9, 10], and answer type validation from the named-entity of the expected answer type.

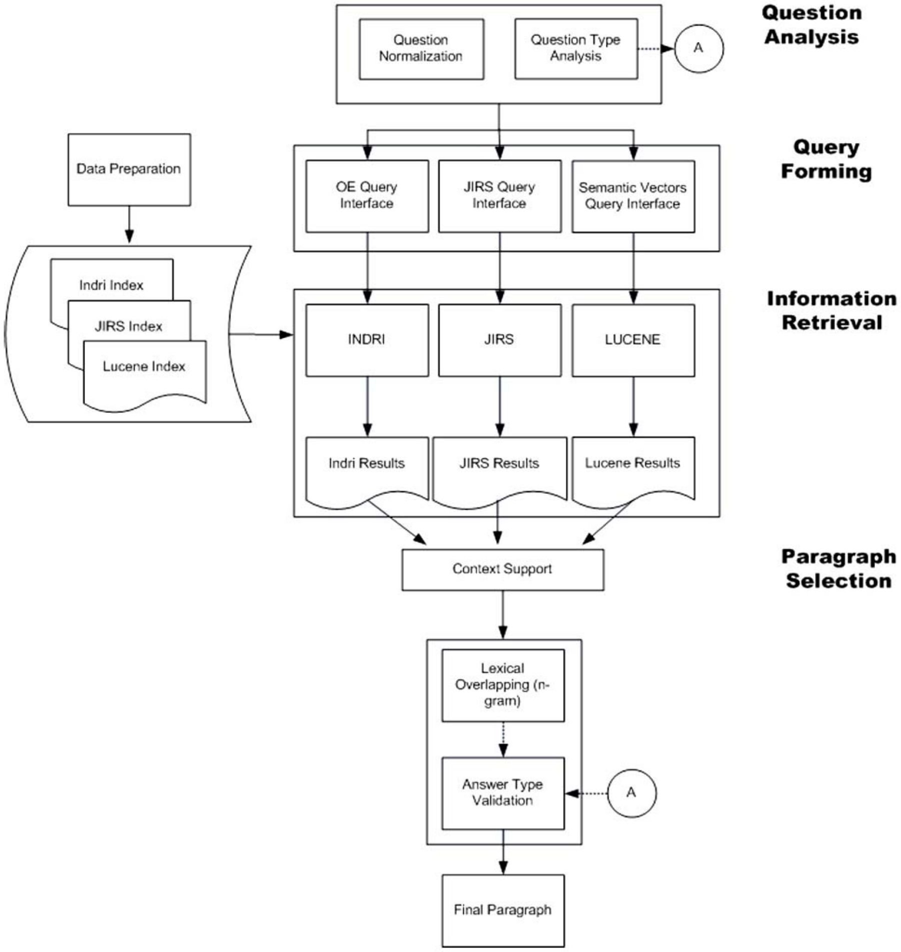

Fig. 1 High Level Architecture of Proposed Approach

3.3 Training Data from ResPubliQA 2009

The above explained architecture was tested using the data from ResPubliQA 2009 for all 500 questions, the results of which can be seen in Table 1.

Table 1 Test Session ResPubliQA 2009

IR Engine	Baseline c@1	+Context	Difference
Indri	29.6	36.4	(+) 6.8
JIRS	38.8	45	(+) 6.2

From this initial experiment, we can see that our proposed approach, as described in previous sub-section, outperformed the baseline accuracy of Indri and JIRS by 6.8% and 6.2% respectively. The graphical interpretation from the above experimental result is presented in Fig. 2.

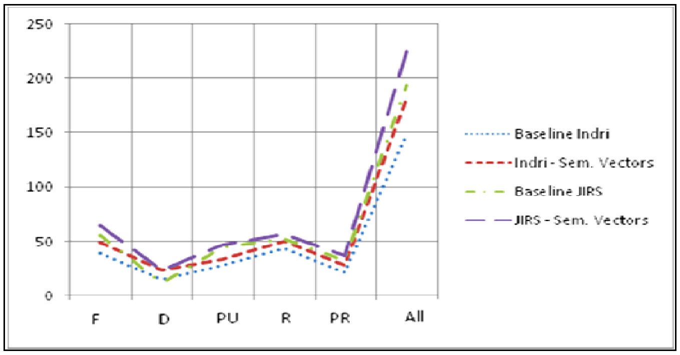

Fig. 2 Graphical Interpretation of ResPubliQA 2009 Test Questions

Fig. 2. gives the number of right answers for each question type, i.e.: factoid (F), definition (D), purpose (PU), reason (R), procedure (PR), and for all question types (All) from last year's ResPubliQA training data. There are a number of conclusions that can be drawn from this initial experiment:

1. JIRS' retrieval engine gives the best result when used as a standalone system.
2. D and PR question types have lower accuracy in comparison with F, PU, and R.
3. The combination of context support and JIRS gives the best accuracy.
4. Semantic Vectors is not suitable as a standalone system for this task, but it can give a supporting 'second opinion' when passages from Indri or JIRS are not supported enough as a final answer.

These initial results increased our confidence that context supporting will improve the overall performance of our architecture. Our final scenarios for the submitted runs in ResPubliQA 2010 will be explained in the next two sub-sections.

3.4 Pattern Based – Context Support

For our first submitted run, we combined the technique of pattern-based question answering, as proposed in [1, 5] with the results from Semantic Vectors. There are five question patterns manually developed for each question type: factoid, definition, reason-purpose, procedure, and opinion. Each question pattern will be responsible for the expected answer type analysis during the paragraph selection phase. An example of the developed patterns is shown in Listing 1, for the opinion question type.

```
QUESTION_TEMPLATE: What is the <CO> opinion of <TO>
  ANSWER_TEMPLATE: Opinion: <PO>
   what (a|the)? <CO> (position|opinion|feelings|ideas) (with respect
   to) <TO>
```

Listing 1 Example of Question Pattern (Opinion)

The detailed strategy of the paragraph selection phase can be seen in Listing 2. In this scenario, the supporting context (the first step in listing 2), is used to select passages that we believed 'share' the same context, and hence they can be used confidently as the final answer. If the result of Indri and Semantic Vectors did not return the same passage, we evaluate the top-5 of Indri results by applying *n*-gram overlapping (step 2.a) If *n*-gram overlapping is unsuccessful, i.e. all passages have the same amount of *n*-gram overlapping, we use a textual containment strategy to calculate the best proportion of query terms that can be found in the passages (step 2.b), and at the same time try to find a named-entity that is expected to be present in the passages (step 2.c). 'No answer' will be returned if there is no passage that meets the conditions in the strategy (step 3). Note that by using containment, passages whose text is too long will be discarded. This step is used essentially to anticipate the definition question type, which usually only has one or two terms from the query that can be found back in the candidate passages.

```
1.  Compare the retrieved passages by Indri and Semantic Vectors. If the
    top-1 returned document is the same, then return the passage offered
    by Indri as final answer.
2.  Otherwise:
    a.  Get top-5 retrieved passages from Indri retrieval, count the
        query-passage n-gram overlapping (unigram, bigram and trigram)
        from each passage, return passage with the biggest total n-gram
        overlapping as final answer.
    b.  If the total count of n-gram overlapping of top-5 passages is
        all the same, then compare the query supporting text in each
        Indri retrieved passages (textual containment). Sort the
        passages by the best containment as candidate answers.
    c.  If the expected answer type is about facts (time, place, number,
        organization), definition (about a term), or opinion (a person
        name or organization), check the result from step 2.a. If a
        named-entity can be found in the first passage, returns it as
        final answer. Otherwise, try to find named-entity in the lower
        rankings.
3.  If there is no passage found that confirm steps 1, or 2, returns No
    Answer (NOA).
```

Listing 2 Detail Strategy of Scenario 1 (Pattern-based – Context Support)

Table 2 gives some examples of the system's decision according to this scenario for the test questions from ResPubliQA 2010.

Table 2 Examples of System's Response using Scenario 1

Quest. #	Top-1 Indri	Top-1 SV Retrieval	Step (see also Listing 2)	Final Paragraph Returned
0008	p_id=13 doc_id=EP_CRE-20090507-EN_cl.xml	EP_CRE-20090507-EN_cl.xml	1	p_id="13" docid="EP_CRE-20090507-EN_cl.xml"
0001	p_id=45 doc_id=EP_CRE-20091008-EN_cl.xml	jrc31989L0552-en.xml	2.a	p_id="45" docid="EP_CRE-20091008-EN_cl.xml"
0026	p_id=13 doc_id=jrc32002D0268-en.xml	jrc31998D0512-en.xml	2.b, 2.c	p_id="26" docid="jrc32006R1583-en.xml"
0027	p_id=164 doc_id=EP_TA-20090114-EN_cl.xml	EP_TA-20080116-EN_cl.xml	3	NOA

As an example from Table 2, the Quest. # 0008 (*Name a purpose of MEDIA Mundus*), Indri and SV have both retrieved the same top-1 document (*EP_CRE-20090507-EN_cl.xml*). According to our strategy, the system should take step 1 in Listing 2 as the final decision for the paragraph returned.

In another example from Table 2, the Quest. # 0027 (*Define children footwear*), Indri and SV have retrieved different documents. The next step is to determine the best *n*-gram overlapping (step 2.a) between the top-5 Indri retrieved passages and the question terms (*children footwear*). Since all of the top-5 Indri retrieved passages have only one term in common with the question (i.e. the term *children*), the system should now compute the value of textual containment [9], to determine how well the *n*-gram overlapping is in comparison with the length of the passage (step 2.b). From last year's training data, we decided to use a minimum threshold value of 0.02 for textual containment. In the case of Quest. # 0027, all of the top-5 retrieved passages have a textual containment value lower than the threshold, hence the system will return 'NOA' as the final answer (step 3).

3.5 *n*-Gram Passage Retrieval – Context Support

For our second submitted run, we combined the results of JIRS and Semantic Vectors to retrieve passages which could contain answer candidates. The detailed strategy of the paragraph selection phase can be seen in Listing 3.

```
1. Compare the retrieved passages from JIRS and Semantic Vectors. If
   the top-1 returned document is the same, then return the passage
   offered by JIRS as the final answer.
2. Otherwise:
   a. Get the top-5 passages from JIRS retrieval, count the query-
      passage n-gram overlapping (unigram, bigram and trigram) from
      each passage, return passage with the biggest total n-gram
      overlapping as the final answer.
```

Listing 3 Detail Strategy of Scenario 2 (*n*-Gram Passage Retrieval – Contexts Support)

```
  b.  If all of the top-5 JIRS passages have the same total count of
      n-gram, then count the n-gram overlapping (unigram, bigram and
      trigram) of the query in each paragraph of the retrieved top-1
      Semantic Vector document, return passage with the biggest n-gram
      overlapping as the final answer.
  c.  If there is more than one paragraph from the SV top-1 document
      have the same total count of n-gram then return paragraph with
      the best textual containment as the final answer.
```

Listing 3 (cont'd) Detail Strategy of Scenario 2 (*n*-Gram Passage Retrieval – Contexts Support)

The first paragraph selection strategy is to compare the top-1 retrieved passage from JIRS and the one retrieved by Semantic Vectors. If both retrieval systems suggested the same document, then we took the passage returned by JIRS as the final answer. Otherwise, we followed the same strategy as scenario 1 to select the best paragraph. But now, we used the top-1 suggested document from Semantic Vectors as the final paragraph decision. Table 3 gives some examples of the system's decision according to this second scenario for the test questions from ResPubliQA 2010. The explanation for Table 3 is similar to the explanation for Table 2 in the previous sub-section.

Table 3 Examples of System's Response using Scenario 2

Quest. #	Top-1 JIRS	Top-1 SV Retrieval	Step (see also Listing 3)	Final Paragraph Returned
0003	p_id=34 doc_id= EP_CRE-20091008-EN_cl.xml	EP_CRE-20091008-EN_cl.xml	1	p_id=34 doc_id= EP_CRE-20091008-EN_cl.xml
0059	p_id=11 doc_id= jrc31988R3498-en.xml	jrc32000R1609-en.xml	2.a	p_id=11 doc_id= jrc31988R3498-en.xml
0012	p_id=268 doc_id= EP_TA-20070201-EN_cl.xml	EP_TA-20080925-EN_cl.xml	2.b	p_id="283" docid="EP_TA-20080925-EN_cl.xml
0017	p_id=1006 doc_id= jrc32006R0865-en.xml	EP_TA-20070710-EN_cl.xml	2.c	p_id="289" docid="EP_TA-20070710-EN_cl.xml

In Scenario 2, there was no strategy developed for the NOA-answer. The reason for this is considering the final decision rule that selects paragraphs from only one document, i.e. the Semantic Vectors top-1 retrieved document - with no textual containment threshold - and thus, there must be one paragraph returned as the final answer.

4 Results and Analysis

Table 4 gives the result summary of our submitted runs. Context support which combines the passage retrieval from Indri and Semantic Vectors (uiir101PSenen) gives better accuracy than the second configuration which uses the combination of

JIRS-Semantic Vectors (uiir102PSenen). These two configurations differ in the way passages are refined during the paragraph selection phase (cf. Section 3).

Table 4 Submitted Runs Results

Run	Ans.	Ans. Right	Ans. Wrong	Unans-wered	Un-ans. Right	Un-ans. Wrong	Un-ans. Empty	Overall Accuracy	c@1
uiir101PSenen	197	143	54	3	0	3	0	0.72	0.73
uiir102PSenen	200	127	73	0	0	0	0	0.64	0.64

If we observe the accuracy of each question type in detail (Table 5), the ‘Definition’ question type gives the lowest accuracy in each configuration, i.e. 0.43 (Scenario 1) and 0.37 (Scenario 2). This result indicates that the ‘Definition’ question type is the most difficult one to be answered, and our strategy is not suitable to answer this type of question properly. The best accuracy in both scenarios is achieved for the ‘Reason-Purpose’ question type, i.e. 0.87 (Scenario 1) and 0.77 (Scenario 2).

Table 5 Question Types Judgments

JUDGMENTS		RIGHT ANS.		WRONG ANS		ACCURACY	
Q. Type	**# of Quest.**	**Sce. 1**	**Sce. 2**	**Sce. 1**	**Sce. 2**	**Sce. 1**	**Sce. 2**
Factoid	69	53	47	16	22	0.77	0.68
Definition	29	13	11	17	19	0.45	0.38
Reason-Purpose	31	27	24	3	6	0.87	0.77
Procedure	38	22	26	16	12	0.58	0.68
Opinion	33	28	19	5	14	0.85	0.58
TOTAL	200	143	127	57	73	0.72	0.64

It is interesting to explore the accuracy of the ‘Procedure’ question type, which achieved 0.58 accuracy in Scenario 1, and 0.68 in Scenario 2. Most of the question types have question patterns and question terms that are almost identical from one question to another; see for example the ‘Opinion’ question pattern in Listing 1. This is also one of the reasons that Scenario 1 performed better than Scenario 2, except for the ‘Procedure’ question type that has richer patterns and more answer variations.

Table 6 Judgments of “Step #1” in each Scenario

Step 1 Judgements	RIGHT ANS.		WRONG ANS.	
Q. Type	**Sce. 1**	**Sce. 2**	**Sce. 1**	**Sce. 2**
Factoid	13	12	3	3
Definition	1	2	6	2
Reason-Purpose	3	4	1	0
Procedure	3	4	4	2
Opinion	4	2	0	3

Table 6 shows the judgments of the system's decision that was performed by step #1 in each scenario (the complete steps can be found in Listing 2 and 3), which is important in our word-space contextual supporting strategy. By using Scenario 1, there were 38 questions (19% of all questions) that conformed to step #1, and 24 of them were judged correctly (0.63 accuracy). By using Scenario 2, there were 34 questions (17%) that matched the rule in this step, and 24 of the answers were judged correctly (0.71 accuracy). We found that the accuracy is acceptable, but the number of questions that have been evaluated by this step is rather low, and needs to be improved.

5 Conclusions and Future Works

In this paper we have described our context supporting approach in passage retrieval question answering for the English monolingual task. Our experimental results showed that in a specific domain, context supporting - as a 'second-opinion' decision system by using random document projection - can improve the performance of the passage retrieval (paragraph selection) component in a QA system.

It would be interesting to conduct a further study in how to relate word context in broader domains, to support open-domain and multilingual question answering. It would also be interesting to investigate the value of reduced dimensionality from a document collection that formed the word space during retrieval. In this experiment we have only used the number of dimensions of 2000, due to technical restrictions. Perhaps, if the value of the dimension is varied, the context supporting strategy could perform better.

References

1. Schlaefer, N., Gieselmann, P., Schaaf, T., and Waibel, A.: A Pattern Learning Approach to Question Answering within the Ephyra Framework. In: LNAI 4188, pp. 687–694, Springer-Verlag, 2006.
2. Buscaldi, D., Rosso, P. , Gómez-Soriano, J.M., Sanchis, E.: Answering questions with an *n*-gram based passage retrieval engine. Journal of Intelligent Information System (82), Springer, 2009.
3. Sahlgren, M.: An introduction to random indexing. In: Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering (TKE), Copenhagen, Denmark, 2005. SICS, Swedish Institute of Computer Science. Available at the Internet: http://www.sics.se/˜mange/papers/RI_intro.pdf. Retrieved in March 2010.
4. Widdows, D., Ferraro, K.: Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application. MAYA Design, University of Pittsburgh, 2009.

5. Schlaefer, N.: Deploying Semantic Resources for Open Domain Question Answering. Diploma Thesis. Language Technologies Institute School of Computer Science Carnegie Mellon University, 2007.
6. Metzler, D.: Indri Retrieval Model Overview. Available at the Internet: http://ciir.cs.umass.edu/ ~metzler/indriretmodel.html. Retrieved in April 2010.
7. Gómez, J. M., Buscaldi, D., Rosso, P., and Sanchis, E.: JIRS Language-independent Passage Retrieval system: A comparative study. In: Proc. 5th int. conf. on natural language processing (ICON), Hyderabad, India, 4–6 January 2007.
8. Kanerva, P., Kristoferson, J., and Holst, A.: Random indexing of text samples for Latent Semantic Analysis. In: L.R. Gleitman and A.K. Josh (eds.), Proc. 22nd Annual Conference of the Cognitive Science Society (U Pennsylvania), p. 1036. Mahwah, New Jersey: Erlbaum, 2000.
9. Broder, A. Z.: On the resemblance and containment of documents. In: Compression and Complexity of Sequences, IEEE Computer Society, 1998.
10. Lyon et al.: Detecting short passages of similar text in large document collections. In: Conference on Empirical Methods in Natural Language (EMNLP2001). pp. 118-125, 2001.
11. Correa, S., Buscaldi, D., Rossio, P.: NLEL-MAAT at CLEF-ResPubliQA. In: Working Notes ResPubliQA 2009.
12. Lopez, Vanessa, et. al.: AquaLog: An Ontology-Portable Question Answering System for the Semantic Web. In: ESWC (European Semantic Web Conference) LNCS (Lecture Notes on Computer Science) 3532, pp. 546 – 562. Springer-Verlag Berlin Heidelberg, 2005.
13. Ittycheriah, Abraham, et. al.: IBM’s Statistical Question Answering System. In: Proceedings of the 10th Text Retrieval Conference (TREC), 2001.
14. Metzler, D. and Croft. W. Bruce: Analysis of Statistical Question Classification for Fact-based Questions. Kluwer Academic Publisher, 2004.
15. Narayanan, S. and Harabagiu, S.: Question Answering Based on Semantic Structures, 2004.
16. Harabagiu, S., et. al.: Employing Two Question Answering Systems in TREC-2005. In: Proceedings of the 14th Text Retrieval Conference (TREC), 2005.
17. Bos, J.: The “La Sapienza” Question Answering System at TREC-2006. In: Proceedings of the 15th Text Retrieval Conference (TREC), 2006.
18. Glöckner, I. and Pelzer, B.: The LogAnswer Project at CLEF 2009. In: Working Notes ResPubliQA 2009.
19. Andrenucci, A. and Sneiders, E.: Automated Question Answering: Review of the Main Approaches. In: Proceedings of the Third International Conference on Information Technology and Applications (ICITA) IEEE Computer Society, 2005.
20. Bilotti, Matthew W. & Nyberg, Eric.: Improving Text Retrieval Precision and Answer Accuracy in Question Answering Systems. In: Proceedings of the ACL 2nd Workshop on Information Retrieval for Question Answering, 2008.
21. Peñas, Anselmo, et. al.: Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. Cross Language Evaluation Forum, 2009.